

## ⑫ 公開特許公報(A)

昭63-284673

⑬ Int. Cl.

G 06 F 15/347  
9/38

識別記号

350

庁内整理番号

H-7056-5B  
A-7361-5B

⑭ 公開 昭和63年(1988)11月21日

審査請求 未請求 発明の数 1 (全10頁)

## ⑮ 発明の名称 情報処理装置

⑯ 特 願 昭62-119728

⑰ 出 願 昭62(1987)5月15日

⑱ 発 明 者 金 澤 敬 東京都港区芝5丁目33番1号 日本電気株式会社内  
⑲ 発 明 者 岡 野 格 山梨県甲府市丸の内1丁目17番14号 甲府日本電気株式会  
社内  
⑳ 出 願 人 日本電気株式会社 東京都港区芝5丁目33番1号  
㉑ 出 願 人 甲府日本電気株式会社 山梨県甲府市丸の内1丁目17番14号  
㉒ 代 理 人 弁理士 内 原 晋

## 明 細 書

の方の命令によるリクエストを抑止する手段とを  
含むことを特徴とする情報処理装置。

## 1. 発明の名称

情報処理装置

## 2. 特許請求の範囲

ベクトルデータを取扱う命令を有し、前記ベクトルデータを転送する複数組のデータバスを有する情報処理装置において、

前記ベクトル命令によるデータ転送を、以上のデータ転送リクエストに展開する展開手段と、

この展開手段で展開されたリクエストの1つに含まれるベクトル要素の全てを含むアドレスの範囲を求める計算手段と、

この計算手段で計算された前記ベクトルデータのアドレス範囲が前記複数組のデータバス間で重複するかどうかを比較する比較手段と、

この比較手段での前記アドレス範囲の比較の結果、重複が認められ、かつ比較対象となったリクエスト間でデータの参照関係があった場合は、後

## 3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は記憶装置とデータバッファ間のデータ転送を行なう情報処理装置に関し、特にロード/ストアを含む多量のデータ転送を参照関係を損なうことなく高速に行なう情報処理装置に関する。

〔従来の技術〕

計算機の高速度化技法として、パイプライン処理は一般的なものである。パイプライン処理は1命令の実行を複数ステージに分割し、命令をステージごとに連続的に流すことにより、1命令の実行時間と実効的に1ステージのサイクルタイムに近づけようとするものである。パイプライン計算機においては、前の命令が終了しないうちに次の命令の実行が開始されるため、データの参照関係の保証が必要となってくる。すなわち、ある命令でメモリからオペランドを読出す場合には、それ以

前の命令によるストアを全て反映したオペランドを読出さなければならない。そのため、オペランドフェッチのアドレスをストアバス内のストア用アドレスと比較し、アドレスが一致すればストアの優先処理、データの差し換え等が行なわれる。

しかし、大量のベクトルデータを高速に処理する必要があるスーパーコンピュータにおいては、上述のような方法による参照関係の保証は性能低下を招くため、ロード/ストア命令の逐次処理、ソフトウェア命令による参照関係の保証等の方法がとられてきた。しかしながら、前者の方式は第10図に示すように、ストア命令が完了するまで、次のロード命令が動作できないため、パイプラインに待ち時間が生じ、データ転送の立上がり時間、すなわち、(最初のベクトル要素が読出されるまでの時間)が増加する。後者の方法は、前者の欠点に加えてソフトウェアの負担が増加するという問題がある。

〔発明が解決しようとする問題点〕

上述した従来のベクトルデータに対する参照関

- 3 -

次に、本発明について図面を参照して詳細に説明する。

第2図を参照すると、本発明の一実施例は、オペランドバッファ101、記憶装置107、これらの装置101および107の間に存在する1本のストア専用パイプライン103、2本のロード専用パイプライン104および105の計3本のデータ転送バスを備えている。

命令は命令バッファ100から読出され、命令制御部102で解読される。解読された命令がベクトル転送命令であれば、記憶制御部106に対してリクエストを出す。この際、命令をデコードして得られた情報も線114~118を通して記憶制御部106に転送される。記憶制御部106は、命令制御部102からリクエストを受付けると、リクエストの種類に応じてデータ転送バスを割付け、各々のパイプラインに対して線119~121を使ってデータ転送を指示する。データ転送要求は図示しない信号によって記憶装置107に対しても出される。記憶装置107は前記3本

- 5 -

本の保証方法にはパイプラインが空いているにもかかわらず、無駄な待ち時間が生ずるという欠点がある。この待ち時間はベクトルの要素数が多くなる程増大する。

〔問題点を解決するための手段〕

本発明の装置は、複数のロード/ストアパイプラインを含む情報処理装置において、

ベクトル命令によるデータ転送を1以上のデータ転送リクエストに展開手段と、

この展開手段で展開されたリクエストの1つに含まれるベクトル要素の全てを含むアドレス範囲を求める計算手段と、

この計算手段で計算された前記ベクトルデータのアドレス範囲が前記複数組のパイプライン間で重複するかどうかを比較する比較手段と、

この比較手段でのアドレス範囲の比較の結果が認められ、かつ比較対象となったリクエスト間でデータの参照関係がある場合は、後の方の命令によるリクエストを抑止する手段とを有している。

〔実施例〕

- 4 -

のパイプライン103~105のデータ転送要求に対して並列に動作可能であり、1回のリクエストに対するデータ転送が終了することにより、各パイプライン毎に線122~124を通してデータ転送の終了を記憶制御部106に報告する。

第2図および第2図の記憶制御部106の構成を詳細に記述した第1図を参照すると、記憶制御部106は本発明の特徴を最もよく表わすもので、第2図のロード/ストアパイプライン103~105で現在処理中のリクエストの参照関係(リクエストの前後、およびアドレス)を管理し、次のリクエストが送出可能であるかどうか判断されている。

本実施例では、第2図の命令制御部102で解読された命令は記憶制御部106中のリクエスト展開部203で最大8つのリクエストに展開される。展開されるリクエストの数は、命令制御部102から線116を通じて与えられる命令の要素数によって左右される。

次に、このリクエスト展開部203を第3図を

- 6 -

参照しながら詳細に説明する。

レジスタ401～403には、各パイプラインに割付けられたリクエストの個数、レジスタ404～406には、各パイプラインのリクエストに対する、第1図の記憶装置107からのリブライの個数のカウント値がセットされる。レジスタ401～403は、パイプラインが割付けられた時に、第2図の命令制御部102から線116を通じて送られてくるリクエストの個数がロードされる。レジスタ404～406は、各パイプラインにリクエストが割付けられた時にリセットされ、その後、第1図の記憶装置107からのリブライが各パイプライン毎に線122～124と通じて来る度に“1”ずつカウントアップされる。比較器416～418では、レジスタ401～403の内容と、レジスタ404～406の内容とを各々比較している。すなわち、リクエストを出して、リブライが全て戻ってきていないパイプラインについては、比較器416～418は不一致を示し、リブライが全て戻ってきたパイプラインでは、比

- 7 -

付けられる。2本のパイプラインがともに使用中の場合の新たなリクエストは第2図の命令制御部106で抑止される(図示せず)。

比較器418に対応する第2図のパイプライン105が空の時にリード系のリクエストがきた場合の動作を以下に示す。比較器418の出力447が論理“1”、バッファ422の正相出力451が論理“1”、リード系のリクエストを示す線118が論理“1”のため、アンドゲート428の出力220は論理“1”になる。

この線220はパイプライン105を割付ける信号を伝送する。

一方、バッファ422の反転出力452は論理“0”となるため、アンドゲート427の出力219は論理“0”になる。線219はパイプライン104を割付ける信号である。次に比較器418に対応するパイプライン105が使用中で比較器417に対応するパイプライン104が空の時にリードリクエストがきた場合の動作を以下に示す。比較器417の出力446は論理“1”、

- 9 -

比較器416～418は一致を示す。つまり、比較器416～418の出力は、パイプラインの使用状況を示している。

第2図の命令制御部102からのリクエストに対するパイプラインの割付けはこの比較器416～418の出力を見て決定される。

命令制御部102からのリクエストは、そのリクエストがロード系か、ストア系かにより線118、または117をそれぞれ有効にする。ストア系リクエストに対しては、ストアパイプラインが1本しかないため、アンドゲート426にて、比較器416の出力と、ストア系リクエストを示す線117の内容との論理積をとり、その結果を線218に出力する。線218の内容は第2図の命令制御部102からのリクエストに対してストアのパイプライン103の割付けを示す。リード系リクエストに対しては、リードパイプラインが2本あるため、どちらか空いた方のパイプラインを割付ける。2本のパイプラインがともに空いている時は、リードパイプライン105が優先的に割

- 8 -

比較器418の出力447は論理“0”、バッファ422の正相出力451は論理“0”、反転出力452は論理“1”、リード系リクエストを示す線118は論理“1”のため、アンドゲート427は論理“1”、アンドゲート428は論理“0”になる。この場合、リクエストはパイプライン104に割付けられる。

以上の方法によってパイプラインの割当てが決定される。ここで、ストアの次にパイプライン103が割当てられた場合の動作について詳細に説明する。レジスタ401には、該当するリクエストの個数が蓄えられる。

これと同時にレジスタ404、および407がリセットされる。これにより、比較器416、および419は不一致を示す。比較器419はレジスタ401および407のそれぞれの内容を比較するが、不一致を検出すると出力448が論理“1”となる。この時、後述するリクエスト有効信号225が論理“1”であれば、アンドゲート423の出力119は論理“1”になる。この信

- 10 -

号119に回答して、ストアパイプライン103はデータの転送を開始し、同時に、レジスタ407の内容を+1する。このリクエストは、レジスタ401の内容とレジスタ407の内容が等しくなるまで繰り返される。この間に信号線225が論理"0"となった時は、この信号が論理"1"になるまでリクエストは抑止される。線119から出力されたリクエストはストアパイプラインを介して記憶装置107に報告される。記憶装置107はリクエストとともにデータ転送を開始し、データ転送が終了するとデータ転送終了報告を線122を介して、リクエスト展開部203に送り返してくる。

リクエスト展開部203では、この場合により、1つのリクエストの処理が終了したことを知り、レジスタ404を"1"だけ加算する。全てのリクエストに対するデータ転送終了報告が線122から送られてくると、レジスタ404の内容はレジスタ401の内容と等しくなるはずである。これにより、比較器416の出力が"1"になり、

-11-

要素数を32個に固定している)を加算回路301~307で加えていく。命令制御部102から線114を介して受取ったリクエストアドレスをS、線115を介して受取った要素間距離をdとすると、各リクエストの先頭アドレスはS、 $(S+32 \times d)$ 、 $(S+32 \times 2 \times d)$ 、…、および $(S+32 \times 7 \times d)$ となる。要素間距離は最大 $2^m$ である。リクエストアドレスのビット数をnビットとすると、上記加算回路301~307で計算されたアドレスのうちの上位 $n-(m+5)$ ビットが線210~217を通じてアドレス比較部202に送られる。

第5図を参照すると、アドレス比較部202では、現在のロード/ストアパイプライン103~105内にあるリクエストの先頭アドレスが全て記憶されている。本実施例では最大8リクエスト/パイプラインである。各要素のリクエストを送出する際は、同一のアドレスをアクセスするコマンドは他のパイプラインにないかどうかをチェックされる。

-13-

パイプラインが空で次のリクエストを受付けられる状態になったことを示す。

リクエストがリードパイプライン104、および105に割付けられた場合も同様の動作が行われる。以上述べたように、リクエスト展開部203では、命令制御部102から受取ったリクエストを記憶装置107に対するリクエストに展開し、パイプラインに割付ける機能がある。

再び第1図を参照すると、リクエスト展開部203で展開されたリクエストに対応するアドレスは、アドレス計算部201で求められ、アドレス比較部202に送られる。第4図を参照すると、アドレス計算部201では、第2図の命令制御部102から線114、および115を介して受取ったリクエストアドレス、および要素間距離から各リクエストの先頭の要素のアドレスを求めている。最初のリクエストの先頭のアドレスは命令制御部102から受取ったリクエストアドレスに等しいが、以降のリクエストの先頭アドレスは、要素間距離 $\times 32$ (本実施例では、1リクエストの

-12-

第5図を参照すると、アドレス比較部202はストアのパイプライン103のためのチェック回路536、リードのパイプライン104のためのチェック回路537、およびリードのパイプライン106のためのチェック回路538を備えている。チェック回路537および538は内部の回路構成の記述が省略してあるが、チェック回路536の回路構成と全く同一である。

まず最初に、第2図の命令制御部102からのリクエストが第1図のリクエスト展開部203でストアパイプライン103に割り付けられた場合を想定してみる。

ストアパイプライン103が割付けられた場合は線218が論理"1"になる。

一方、第1図のアドレス計算部201では各リクエストに対するリクエストアドレスが計算され、上位 $n-(m+5)$ ビットが線210~217と通じて送られている。アドレス比較部202では、この各リクエストのリクエストアドレスをレジスタ501~508に書き込む。同時に有効な要素

-14-

数に応じてレジスタ509～516にも“1”がセットされるレジスタ509～516に“1”をセットしたレジスタが以下に述べるアドレス比較の対称となる。

次に、前記手段により展開されたリクエストの1つを送出する際のアドレス比較の様子を示す。リードパイプライン104内のリクエストを送出する場合は、第5図のチェック回路537中の相当するレジスタ(チェック回路536中のレジスタ501～508に相当する)の中から送出的リクエストに対応するアドレスがチェック回路537中の相当するレジスタ(チェック回路536中のレジスタ535に相当する)に読出される。前記レジスタに読出されたアドレスは線583を介して比較回路517～524に入力される。比較回路517～524では、レジスタ501のアドレスと前記線583から入力されたアドレスが比較される。このアドレスは、リクエストの最初の要素のアドレスの上位 $m+5$ ビットであり、リクエスト内の全ての要素はその上位 $m$

-15-

第1図の追越しチェック部204に出力される。

第1図および第1図の追越しチェック部204の詳細な構成を示す第6図を参照すると、追越しチェック部204では、各パイプライン間のリクエスト順位、アドレスの比較の結果から、データの参照関係を乱すリクエストの送出手を検出し、抑止する。

第2図の命令制御部102からリクエストを受取った際に、そのリクエストの処理順序を記憶しておくため、割当てたパイプラインに応じてレジスタ601および602をセットまたはリセットする。リクエストの処理順序はストア系→リード系→1, 2について判断できれば十分のため、レジスタは2ビットである。

ストア系リクエストにストアパイプライン103を割当てた場合は、線218が論理“1”になり、2ビットのレジスタ601および602はともに“1”にセットされる。

リード系リクエストにリードパイプライン104を割当てた場合は、線219が“1”になり、レ

( $m+5$ )ビットアドレスが前記アドレス±1に含まれる。従って、比較回路517～524では、アドレスの上位 $m+5$ ビットが比較され、差が±2以内であれば一致したものと判断される。

前記方法により、重複されたアドレスが比較回路517に入力されており、かつ有効フラグ509に“1”がセットされていれば、比較回路517の出力566が“1”になる。この信号566はオア回路533で論理和がとられ、出力221を“1”にする。これは他のレジスタ502～508でアドレスの一致が検出された場合も同様である。

以上のアドレスの比較対はストアパイプライン103対ロードパイプライン104、およびその逆、ストアパイプライン103対ロードパイプライン105、及びその逆の計4通りがある。ロードのパイプライン同士の比較がないのは、ロード間では追越しが起こっても問題にならないからである。

以上の比較結果は線221～224を通じて、

-16-

レジスタ601が“0”にリセットされる。リードパイプライン105が割当てられた場合は、線220が“1”になり、レジスタ602が“0”にリセットされる。以上の制御により、レジスタ601は、その内容が“1”ならばロードよりもストアが優先、“0”ならばストアよりもロードが優先であることを示す。レジスタ602についても同様である。

以上のリクエストの前後関係、およびアドレス一致情報から、追越しが可能かどうかの判断が行われる。

線223は第2図のストアパイプライン103からのリクエストに対して、リードパイプライン104に同一のアドレスがあるかどうかの比較結果が伝送される。ストアリクエストがリードリクエストよりも優先で、すなわちレジスタ601の内容が論理“1”で、かつアドレス一致を検出していれば、追越しが起こるため、このストアリクエストを抑止しなければならない。アンドゲート603では線223と、レジスタ601の正相出

-18-

-17-

方620を入力とし、線624に出力している。  
この出力624が論理0の時はオアゲート607  
の出力225（反転出力）は論理“0”となり、  
前述したリクエスト展開部203に送られ、リ  
クエストが抑止される。

線224は同様に、第2図のストアパイプライン103からのリクエストに対してリードパイプライン105に同一のアドレスがあるかどうかの比較結果である。この場合はレジスタ602の内容を参照して、追越しが起これば前記のケースと同様にストアリクエストが抑止されている。

線221は、第2図のリードパイプライン104からのリクエストに対して、ストアパイプライン103に同一のアドレスがあるかどうかの比較結果である。

リードリクエストがストアリクエストよりも後発で、すなわちレジスタ601の内容が論理“0”であり、かつアドレス一致を検出していれば追越しが起これるため、このリードリクエストを抑止しなければならない。アンドゲート604では、線

-19-

レスで比較していたが、第2の実施例では、1つのリクエストに対するベクトル要素の始点と終点を求め、このベクトル要素の範囲で比較する。

第2の実施例のアドレスの比較回路の部分抜き書きした第7図を参照すると、レジスタ702および703が第5図のレジスタ501に相当する。レジスタ702は始点ポインタ、レジスタ703は終点ポインタを格納する。一方、第5図で与えられる比較アドレス583は線713および714に相当する。線713は始点ポインタ、線714は終点ポインタをそれぞれ伝送する。ここで、2つの閉領域（一次元）の重複には計4種類のモードがある。

第8図を参照すると、(1)のモードは第7図のアンドゲート708に、(2)のモードは第7図のアンドゲート711に、(3)のモードは第7図のアンドゲート709に、(4)のモードは第7図のアンドゲート710にそれぞれ相当する。比較回路704～707は大小比較の結果を出力している。オアゲート712は前記4つのモードの論理和をとり、

-21-

221とレジスタ601の反転出力621を入力を入力とし、線226に反転出力される。この信号226が“0”の時は前述のリクエスト展開部203でリードパイプライン104に対するリクエストが抑止される。

線222は同様に、リードパイプライン105からのリクエストに対して、ストアパイプライン103に同一のアドレスがあるかどうかの比較結果である。この場合はレジスタ602の内容を参照して、追越しが起これば前記のケースと同様にリードパイプライン105に対するリクエストが抑止される。

前記第1の実施例では、アドレス範囲の比較回路を簡略化するために最初のベクトル要素と最後のベクトル要素を含む上位何ビットかのアドレスでのみ比較したため、実際にアドレス範囲が重複していなくても重複を示すことがあった。

第2の実施例では、第1の実施例のアドレス比較部202の構成を多少変更する。

第1の実施例では、ベクトル要素の単一のアド

-20-

出力727を出力している。出力727は第5図の線566に相当する。

以上のような比較回路を第5図の比較回路と置きかえることにより、より厳密なアドレス比較が可能になる。

#### 〔発明の効果〕

以上説明したように本発明は、データの参照関係を保障するためのパイプラインの待ち時間を短縮できるという効果がある。

従来の方法によるパイプライン待ち合わせを示す第10図を参照すると、A番地に対するベクトルストア命令の直後に同じA番地からデータを読む場合は、データの参照関係の保障が必要である。従来の方法では、ロード/ストアの逐次処理、ソフトウェア命令による待ち合わせを行っていたが、この方法では、前のベクトルストア命令が完全に終了するまで次のロード命令は起動されないため、待ち時間が生じてしまう。この待ち時間はベクトルの要素数が増加するほど大きくなる。これに対して本発明の方法によるパイプライン待ち合わせを

-22-

示す第9図を参照すると、本発明の方法によれば、1つのベクトルロード/ストア命令は数個のメモリアクセリクエストに展開され、各々のリクエストの単位でデータの参照関係がチェックされるため、パイプラインの待合わせ時間を、展開されたリクエストの処理時間程度に減らすことができる。

#### 4. 図面の簡単な説明

第1図は本発明の一実施例を示す図、第2図はデータ転送系を抜き出した情報処理装置の構成を示す図、第3図は第1図のリクエスト展開部203の詳細な構成を示す図、第4図は第1図のアドレス計算部201の詳細な構成を示す図、第5図は第1図のアドレス比較部202の詳細な構成を示す図、第6図は第1図の追越しチェック部204の詳細な構成を示す図、第7図は本発明の第2の実施例のアドレス比較回路の構成を示す図、第8図は、第7図のアドレス比較回路の動作を説明するための説明図、第9図は本発明の効果を説明す

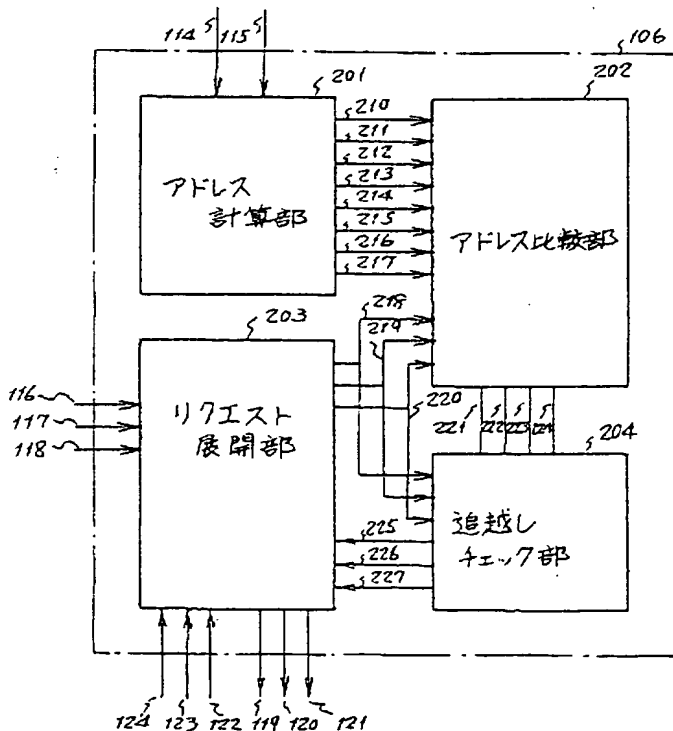
-23-

るための図、および第10図は本発明と従来の方法を比較するための図である。

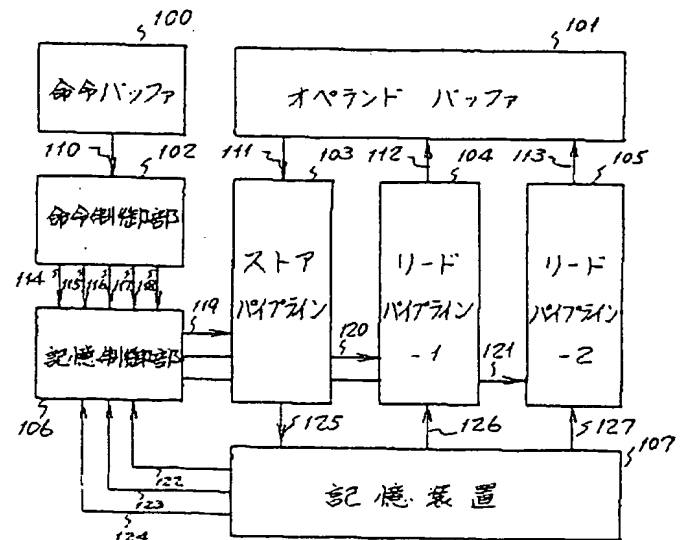
第1図から第10図において、100……命令バッファ、101……オペランドバッファ、102……命令制御部、103……ストアパイプライン、104、105……リードパイプライン、106……記憶制御部、107……記憶装置、201……アドレス計算部、202……アドレス比較部、203……リクエスト展開部、204……追越しチェック部、301～307……加算回路、401～409……レジスタ、410～415……+1加算回路、416～421……比較回路、422……反転出力付きバッファ、423～428……アンド回路、501～516……レジスタ、517～532……比較回路、533～534……オア回路、535……セレクト付きレジスタ、601～602……レジスタ、603～606……アンド回路、607……オア回路、702～703……レジスタ、704～707……比較回路、708～711……アンド回路、712……オア回路。

代理人 弁理士 内 原

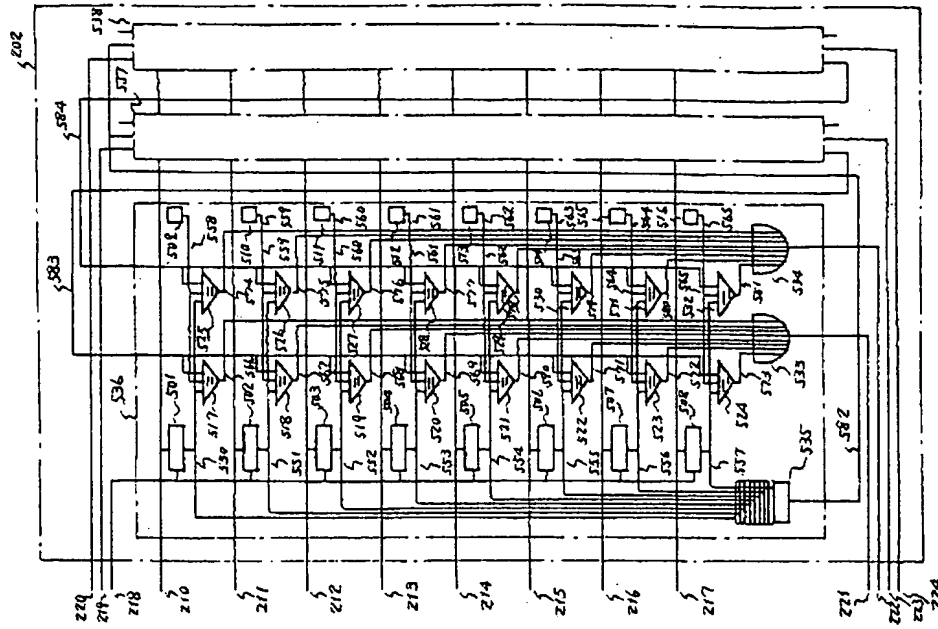
-24-



第1図

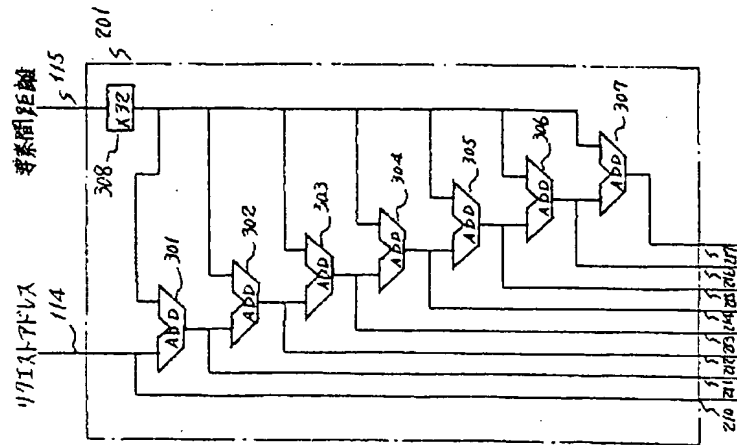


第2図



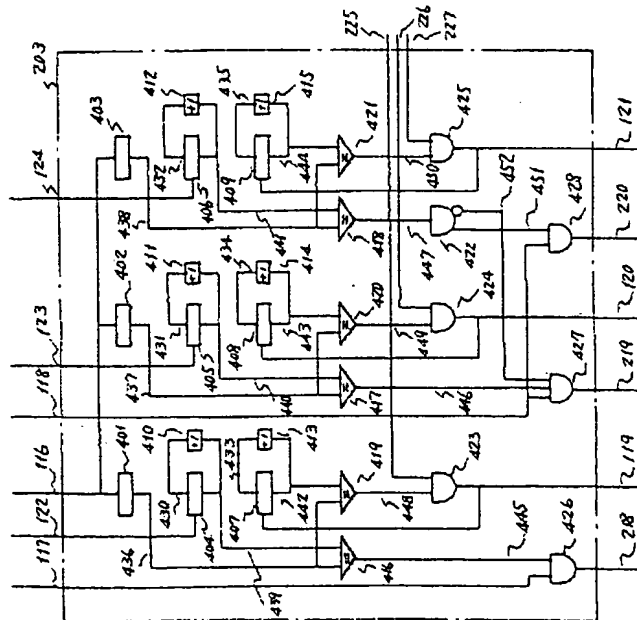
501~516 : レジスタ  
517~532 : 比較回路  
533~534 : OR回路  
535 : ゼロ検出レジスタ

第 5 図



301~307 : 加算回路  
308 : 掛算回路

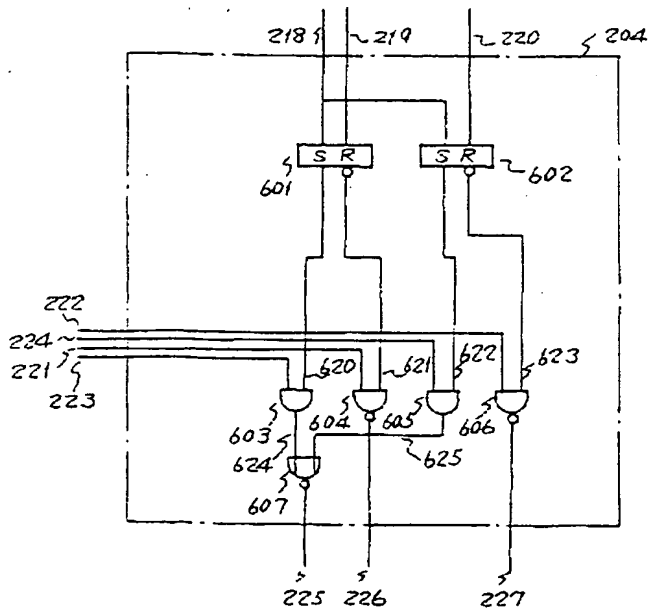
第 4 図



401~409 : レジスタ  
410~415 : +1加算回路  
416~418 : 比較回路(一致の場合のみ出力)  
419~421 : 比較回路(不一致の場合のみ出力)  
422 : 反転出力付きハーフア  
423~428 : AND回路

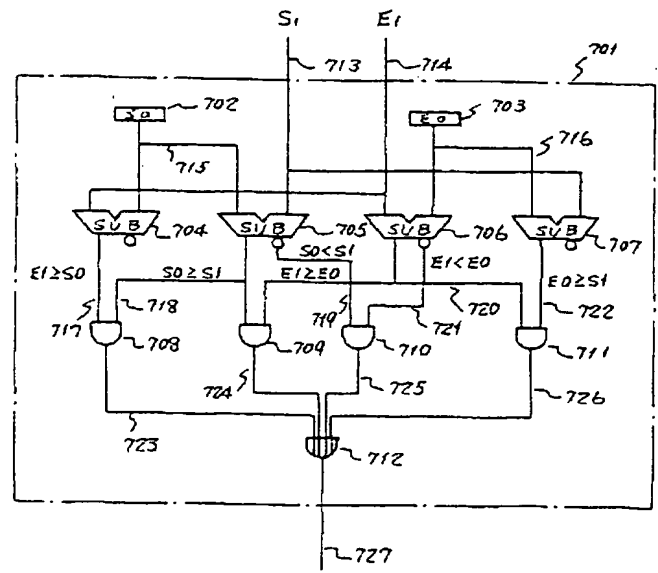
第 3 図





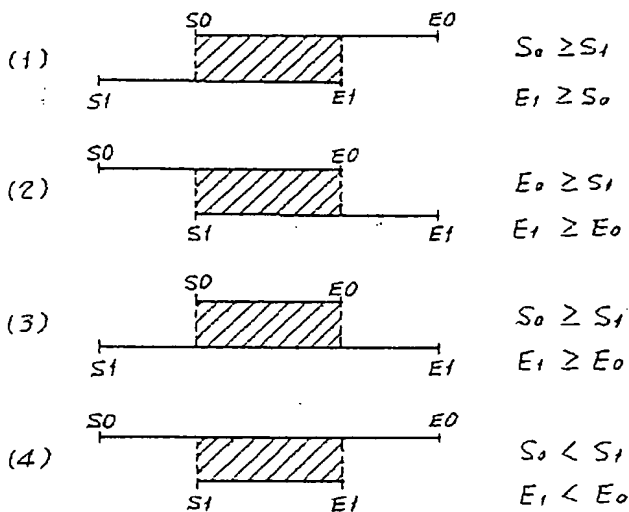
601~602: レジスタ  
603~606: AND回路(一部転出力)  
607: OR回路(転出力)

第 6 図

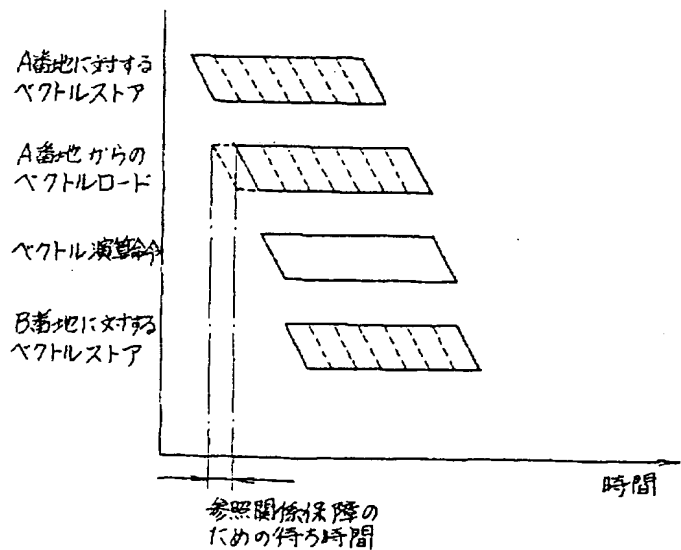


702~703: レジスタ  
704~707: 比較回路(大小比較)  
708~711: AND回路  
712: OR回路

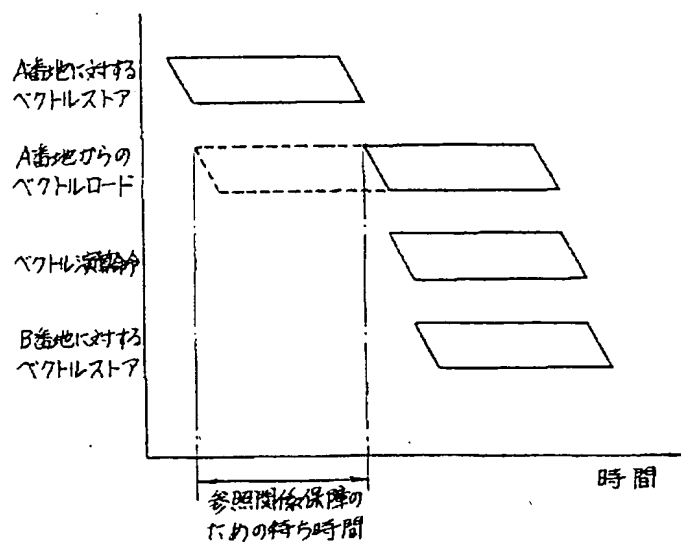
第 7 図



第 8 図



第 9 図



第 10 図